UDC 004.942:001.57

THE METHOD OF HYDRODYNAMIC MODELING USING A CONVOLUTIONAL NEURAL NETWORK

Novotarskyi M. A. – Dr. Sc., Professor of the Department of Computer Engineering, National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

Kuzmych V. A. – Post-graduate student of the Department of Computer Engineering, National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

ABSTRACT

Context. Solving hydrodynamic problems is associated with high computational complexity and therefore requires considerable computing resources and time. The proposed approach makes it possible to significantly reduce the time for solving such problems by applying a combination of two improved modeling methods.

Objective. The goal is to create a comprehensive hydrodynamic modeling method that requires significantly less time to determine the dynamics of the velocity field by using the modified lattice Boltzmann method and the pressure distribution by using a convolutional neural network.

Method. A method of hydrodynamic modeling is proposed, which realizes the synergistic effect arising from the combination of the improved lattice Boltzmann method and a convolutional neural network with a specially adapted structure. The essence of the method consists of implementing a sequence of iterations, each of which simulates the process of changing parameters when moving to the next time layer. Each iteration includes a predictor step and a corrector step. At the predictor step, the lattice Boltzmann method works, which allows us to obtain the field of fluid velocities in the working area at the next time layer using the field of velocities at the previous layer. At the corrector step, we apply an improved convolutional neural network trained on a previously created data set. Using a neural network allows us to determine the pressure distribution on a new time layer with a predetermined accuracy. After adding the fluid compressibility correction on the new time layer, we get a refined value of the velocity field, which can be used as initial data for applying the lattice Boltzmann method at the next iteration. Calculations stop when the specified number of iterations is reached.

Results. The operation of the proposed method was studied on the example of modeling fluid movement in a fragment of the human gastrointestinal tract. The simulation results showed that the time spent implementing the simulation process was reduced by 6–7 times while maintaining acceptable accuracy for practical tasks.

Conclusions. The proposed hydrodynamic modeling method with a convolutional neural network and the lattice Boltzmann method significantly reduces the time and computing resources required to implement the modeling process in areas with complex geometry. Further development of this method will make it possible to implement real-time hydrodynamic modeling in three-dimensional domains.

KEYWORDS: hydrodynamic modeling, convolutional neural network, lattice Boltzmann method.

ABBREVIATIONS

LBM is a lattice Boltzmann method;

CNN is a convolutional neural network;

BGK is a Bhatnagar-Gross-Krook model.

NOMENCLATURE

f is a distribution function;

x is a vector that specifies the position of the elementary volume of liquid in space;

k is a discretization index of the kinematic velocity;

v_k is a kinematic velocity vector;

w_k is weight factor for equilibrium function;

t is a parameter that specifies a point in time;

f^{eq} is an equilibrium distribution function;

η is a kinematic viscosity;

τ is a time of relaxation;

Δx is a grid spacing;

Δt is a time step;

c is a speed in the grid;

$\bar{\tau}$ is a viscous stress tensor;

$\otimes$ is the Kronecker product;

φ is a body forces;

μ is a dynamic viscosity;

$\bar{I}$ is an identity tensor.

INTRODUCTION

The rapid growth in the popularity of artificial neural networks, methods of analyzing large volumes of data, and other alternative approaches has led to the discovery of several breakthrough technologies. In particular, significant progress is observed in studying complex physical processes that can be mathematically described by boundary value problems based on differential equations with partial derivatives. In the paper, we will consider applying this approach to determining the hydrodynamic parameters of liquids. The traditional way of modeling the corresponding physical process is the numerical solution of the boundary value problem, which includes the flow continuity equation and the Navier-Stokes equation. In the practical implementation of this approach, in the case of a modeling area with a complex shape, some difficulties always arise, which are manifested due to the problems of achieving convergence of the corresponding numerical method, which is inextricably linked with the need to use significant computing resources to ensure obtaining results with high accuracy. However, the high accuracy of

DOI 10.15588/1607-3274-2023-4-6

the results could be more accurate in many cases due to the impossibility of accurately determining the initial data for solving this or that boundary value problem. A natural approach to solving this problem is developing a toolkit to solve the given situation with some predetermined approximation.

The object of study is the process of modeling the physical phenomenon of the movement of liquids in objects with complex geometry under the influence of external forces.

The subject of the study The subject of the study is a modeling method that uses the synergistic effect of the joint use of the improved lattice Boltzmann method and the application of a convolutional neural network with a special structure.

The purpose of the work The purpose of the work is to shorten the time of modeling changes in fluid movement parameters and increase the accuracy of parameter determination by correcting the deviation from compressibility.

1 PROBLEM STATEMENT

In the general case, modeling of fluid movement is performed by solving a boundary value problem based on the Navier-Stokes equation. For this purpose, the discretization of the equation and domain is used to apply the appropriate numerical methods. The main problem that arises on this path is ensuring the convergence of numerical methods in the case of studying an area with complex geometry and the significant computational complexity of the corresponding algorithms, which leads to a considerable expenditure of time and computing resources.

An alternative approach is to use the lattice Boltzmann method. This method describes the movement of the liquid at the mesoscopic level through the interaction of the elementary volumes of the liquid, which is represented by the Boltzmann equation:

$$\frac{\partial f}{\partial t}+v\frac{\partial f}{\partial x}=\frac{f-f^{eq}}{\tau}. \qquad (1)$$

The right-hand side of this equation describes the collision process of elementary volumes. This work uses the so-called BGK model, which is often applied to liquids moving at a speed that does not exceed the Mach number for the given liquid. The disadvantage of this approach is the accumulation of errors when determining the hydrodynamic parameters due to density fluctuations.

To overcome the abovementioned shortcoming, we offer a method of specifying the fluid pressure on each time layer.

2 REVIEW OF THE LITERATURE

Applying the lattice Boltzmann method makes it possible to describe the dynamics of the velocity field by colliding elementary volumes of liquid at the mesoscopic level. The history of publications on LBM already dates back several decades[1]. Many practical applications with a severe theoretical basis were implemented [2]. The popularity of this method is due to a large number of advantages, which include relatively low computational complexity [3], the possibility of studying areas with complex geometry [4], and the possibility of parallelizing computational processes [5].

A significant drawback of this method is the increase in the error in determining the parameters of the velocity field with an increase in the number of iterations. This shortcoming does not allow modeling complex physical processes with a given accuracy on long time intervals, which are essential for describing operations in living nature.

A neural network is proposed as a tool to overcome this drawback.

The main advantages of using methods based on neural networks:

1. A neural network provides an analytically presented solution that can be applied repeatedly after preliminary training.
2. Methods based on neural networks can be applied to a broad class of differential equations.
3. These methods include a smaller set of parameters and therefore require fewer resources for implementation.
4. Have mechanisms for regulating the level of generalization of the solution.

Today, a relatively large number of varieties of artificial neural networks are known. Most of them correspond to already well-developed methods of solving differential equations. Historically, an approach focused on deep neural networks was one of the first to be proposed [7]. The application of a deep neural network with extended learning based on the backpropagation algorithm is also offered in [8]. The mentioned methods can be used to solve both ordinary differential equations and differential equations with partial derivatives. They can be used to explore two-dimensional and three-dimensional areas. The disadvantage of the methods is the high labor intensity at the neural network learning stage.

Reducing the complexity can be achieved by modifying the structure of the neural network by adding radial basis functions (RBF) and other facilities to the deep neural network [8]. For example, [9] describes a method that combines a deep neural network with an evolutionary algorithm. The paper shows that the proposed structure can approximate arbitrary functions and their derivatives.

The paper [10] proposed a combination of an artificial neural network with the Karhunen-Loev decomposition [11], which is focused on modeling the solution of the two-dimensional Navier-Stokes equation.

Cellular neural networks [10, 11], an analog computing paradigm, are considered alternative structures for implementing differential equation-solving processes. This concept of analog computing is used to solve complex nonlinear differential equations. This method is based on the Taylor series expansion.

The modern approach to solving partial differential equations is based on convolutional neural networks

DOI 10.15588/1607-3274-2023-4-6

(CNN). Compared to deep neural networks, we get a significantly shorter training time and higher accuracy of solution modeling [12]. At the same time, the main significant advantage of using CNN is the possibility of using complex boundary conditions since the geometry of the area can be included in the learning process.

In this work, the use of a convolutional neural network is proposed. The structure of the network and the method are adapted to solving boundary value problems based on the two-dimensional Poisson equation, which is an essential component of the complex way that models the process of movement of an incompressible fluid.

3 MATERIALS AND METHODS

The way of modeling the movement of fluids aims to determine the distribution of pressure and the velocity field in a given area and changes in these parameters over time. The proposed complex method is represented by a cyclically repeated sequence of iterations, each containing two steps: a predictor step and a corrector step. Fig. 1 shows the general algorithm of this method, in which the functions of the predictor are concentrated in block 2, and the corrector's functions are implemented by blocks 4 and 5.

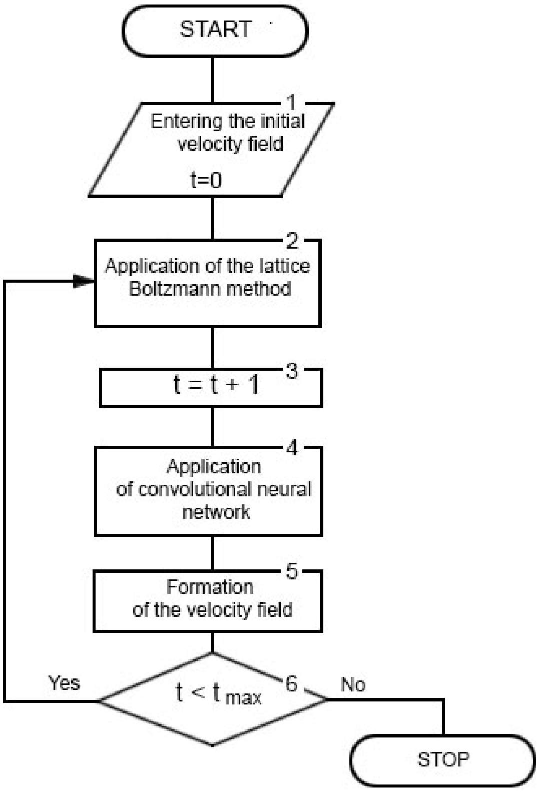
START
1
Entering the initial velocity field
t=0
2
Application of the lattice Boltzmann method
3
t = t + 1
4
Application of convolutional neural network
5
Formation of the velocity field
6
t < t max
Yes
No
STOP

Figure 1 – Algorithm for implementing the complex method

Block 1. Within the framework of this block, the study area is discretized, and the initial data set is formed, which contains the parameters of the velocity field vectors specified in the discretization nodes. Block 2 uses the generated set as initial data to implement the lattice Boltzmann method. The lattice Boltzmann method is used to model the evolution of the parameters of the initial velocity field. Using the value of the velocity field for the time layer to perform one iteration within the framework of this model, we get an updated velocity field on the time layer. We determine the fluid pressure in the discretization nodes on the updated time layer with the value formed in block 3. The pressure distribution is calculated by a previously trained convolutional neural network, represented by block 4. Using the refined pressure value on the time layer in block 5, we determine the corrected velocity field on this time layer. If the current time layer is smaller than the maximum specified time layer, we use the refined velocity field as initial data for the lattice Boltzmann method.

This method is based on the use of the physical model of the Boltzmann equation, which corresponds to the behavior of the liquid flow at the macroscopic level [13]. To apply the Boltzmann equation, it is necessary to consider the fluid flow as a set of elementary volumes. Then the Boltzmann equation describes the evolution of one elementary volume in the form of a distribution function, where is the vector that specifies the position of the elementary volume of liquid in space, is the velocity vector of the elementary volume, and is the parameter that specifies the moment of time. In its general form, the Boltzmann equation can be represented by the expression (1).

We discretize the space of velocities at the mesoscopic level by applying a two-dimensional grid using nine bounce directions of elementary volumes, as shown in Fig. 2.

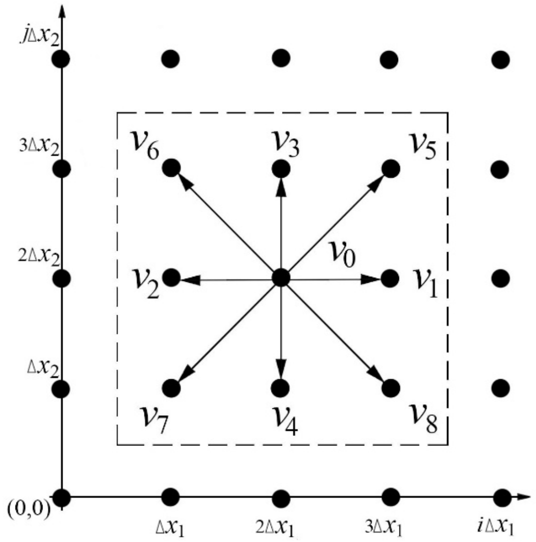

Figure 2 – Discretization of the domain and velocity field

Such a discretization scheme is called D2Q9. It provides sufficient modeling accuracy and is recommended, for example, in [16] for a similar class of problems. If we set $\Delta x = \Delta x_1 = \Delta x_2 = 1$, and the coordinates of the initial node $(0,0)$, then the coordinates of 9 vectors can be written as follows:

$$v_0 = (0{,}0), \quad v_1 = (1{,}0), \qquad v_2 = (-1{,}0),$$

DOI 10.15588/1607-3274-2023-4-6

$$v_3=(0,1),\quad v_4=(0,-1),\quad v_5=(1,1),$$
$$v_6=(-1,1),\ v_7=(-1,-1),\ v_8=(1,-1).$$

For the region discretized in this way, we set the discretized Boltzmann equation:

$$\frac{\partial f_k}{\partial t}+v_k\frac{\partial f_k}{\partial x}=\frac{f_k-f_k^{eq}}{\tau}. \qquad (2)$$

The right-hand side of this equation describes the collision process of elementary volumes. This work uses the BGK model, which is often applied to liquids moving at a speed that does not exceed the Mach number for the given liquid.

Calculating the macroscopic parameters of speed and pressure for a liquid can be calculated as the result of the sequential execution of two steps for each of the discretization vectors v_k. In the first step, we will form a discrete kinetic equation for the distribution function of f_k:

$$f_k(x,t+\Delta t)=f_k(x,t)-\frac{1}{\tau}\left[f_k(x,t)-f_k^{eq}(x,t)\right], \qquad (3)$$

where $\tau=\frac{\eta}{c_s^2}+\frac{1}{2}$, $\Delta t=1$.This equation determines the value of each distribution function at the domain discretization node after colliding the elementary fluid volumes at the corresponding node.

The second step determines the redistribution of the values of the distribution functions on the new time layer:

$$f_k(x+v_k\Delta t,\ t+\Delta t)=f_k(x,t+\Delta t). \qquad (4)$$

After determining the distribution functions at the given time level, we calculate the density and velocity of the fluid in the given discretization node using the formulas:

$$\rho=\sum_{k=0}^{8}f_k,\ u=\frac{1}{\rho}\sum_{k=0}^{8}f_kv_k\ . \qquad (5)$$

Equation (3) also includes the equilibrium distribution function $f_k^{eq}(x,t)$. We determine this function based on the calculations presented in [17]:

$$f_k^{eq}=\rho w_k\left(1+\frac{3}{c^2}v_ku+\frac{9}{2c^4}(v_ku)^2-\frac{3}{2c^2}u^2\right). \qquad (6)$$

where $c=\frac{\Delta x}{\Delta t}$, $w_0=\frac{4}{9}$, $w_1=w_2=w_3=w_4=\frac{1}{9}$, $w_5=w_6=w_7=w_8=\frac{1}{36}$.

Based on [18], we can claim that the method remains stable under $\tau>0.5$. But practical applications of the technique have shown that instability can also occur when tau goes to 0.5. Avoiding a significant growth of the lattice velocity helps maintain the method's stability. In the framework $c_s=\frac{c}{\sqrt{3}}$ the value of pressure is $p=\rho c_s^2$ for the ideal gas. To calculate the pressure in a region filled with a moving incompressible fluid, we must solve a boundary value problem based on the Poisson equation.

Block 4 of the complex method implementation algorithm (Fig. 1) includes solving the Poisson equation separately on each time layer to refine the pressure distribution in the liquid. We use the previously obtained velocity field on this time layer to do this. The transition to the next time layer at low values of the Mach number is performed using the lattice Boltzmann method, which allows us to simulate the change in the velocity field. Since the lattice Boltzmann method, when modeling the parameters of an incompressible fluid, can allow certain density fluctuations that affect the accuracy of determining the velocity field, we must consider these errors when creating the corresponding Poisson equation.

We modify the design scheme proposed in [19] to do this. This scheme consists of three steps. In the first step, the velocity field values are calculated using an explicit iterative scheme, which is based on a discrete representation of the convective and diffuse parts of the momentum equation. In the second step, the resulting velocity field is corrected by the pressure gradient determined after solving the Poisson equation for pressure. The third step is a correction step that ensures the convergence of the iterative determination of the velocity field in the first step.

The main difference of the approach proposed in this paper is that the determination of the velocity field in the next time layer occurs by using the modified lattice Boltzmann method, which is based on the strictly proven fact that at small values of the Mach number, we obtain a result that coincides with the solution in terms of the boundary value problem:

$$\begin{cases}\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho u)=0,\\ \frac{\partial\rho u}{\partial t}+\nabla\cdot(u\otimes u)=\frac{\nabla p}{\rho}+\frac{1}{\rho}\nabla\cdot(\bar{\tau})+\varphi.\end{cases} \qquad (7)$$

The viscous stress tensor for a compressible Newtonian fluid can be determined through the fluid velocity field as:

$$\bar{\tau}=2\mu\left[\frac{\nabla u+(\nabla u)^T}{2}-\frac{1}{3}(\nabla u)\bar{I}\right]. \qquad (8)$$

For a small Mach number when moving to an incompressible boundary value problem, the value of the term can be simplified as follows [20]:

$$\frac{1}{\rho}\nabla\cdot(\bar{\tau})=v\Delta u\ . \qquad (9)$$

© Novotarskyi M. A., Kuzmych V. A., 2023
DOI 10.15588/1607-3274-2023-4-6

To determine the pressure distribution on the new time layer, we define the Poisson equation for pressure [20]:

$$\nabla\cdot\left(\frac{\Delta t}{\rho}\nabla p\right)=\nabla u^{*}, \qquad (10)$$

where $\Delta x_1=\Delta x_2=\Delta x,\ \Delta t=1$ are steps in spatial and temporal coordinates.

Modeling the solution of equation (10) takes place in two stages. The first step is to create a dataset for further use of the convolutional neural network.

We discretize the left and right parts of equation (10). For the left side of equation (10), we get:

$$\nabla\left(\frac{1}{\rho}\nabla p\right)=\frac{p_{i-1,j}-2p_{i,j}+p_{i+1,j}}{\Delta x_1^2}+$$
$$+\frac{p_{i,j-1}-2p_{i,j}+p_{i,j+1}}{\Delta x_2^2}.$$

Due to the fact that we neglect the compressibility of the investigated fluids, we can apply a linear discretization of the right-hand side of equation (10) using the central difference. Then:

$$\frac{\partial}{\partial t}\left(\frac{\partial\tilde{u}_{i,j}}{\partial x}\right)=\left(\frac{\tilde{u}_{i+1,j}-\tilde{u}_{i-1,j}}{2\Delta x_1}+\frac{\tilde{u}_{i,j+1}-\tilde{u}_{i,j-1}}{2\Delta x_2}\right)=S_{i,j}.$$

The resulting system of linear algebraic equations is solved using the iterative Jacobi method [13].

$$p_{i,j}^{n+1}=\frac{\left(p_{i-1,j}^{n}+p_{i+1,j}^{n}\right)\Delta x^2+\left(p_{i,j-1}^{n}+p_{i,j+1}^{n}\right)\Delta x^2}{2\left(\Delta x^2+\Delta x^2\right)}-$$
$$-\frac{\Delta x^2\Delta x^2 S_{i,j}}{2\left(\Delta x^2+\Delta x^2\right)}=$$
$$=\frac{1}{4}\left(p_{i-1,j}^{n}+p_{i+1,j}^{n}+p_{i,j+1}^{n}+p_{i,j-1}^{n}-\Delta x^2 S_{i,j}\right)$$

The obtained pressure distribution is considered and used as an object of the training data set. Training of a convolutional neural network takes place on the training data set formed in this way to solve this type of problem effectively. All problems of obtaining the pressure distribution on each time layer when solving applied problems will be solved in the future with the help of a convolutional network in a significantly shorter time.

After obtaining the values of the pressure field, we adjust the values of the velocity field:

$$u_{i,j}=\tilde{u}_{i,j}-\frac{1}{\rho_{i,j}}\left(\frac{p_{i+1,j}-p_{i-1,j}+p_{i,j+1}-p_{i,j-1}}{2\Delta x}\right).$$

Thus, we get the velocity field on the new time layer. We use these data as initial data when applying the Boltzmann lattice model. This process is repeated if the simulation process continues in time.

Let us briefly consider the development of the mentioned convolutional network and the results of its application.

Based on the previous work [21], a convolutional neural network was developed to model the solution of the boundary value problem based on the Poisson equation. Some changes were made to adapt the basic structure of the neural network to the actual problem. The general structure of the developed neural network is shown in Fig. 3.

First, the size of the input data was increased – from 96×96 to 128×128, which made it possible to simulate the movement of liquids with greater accuracy. The “bottom-up pathway” and “top-down pathway” parts of the network were also expanded by adding additional convolutional layer blocks. Also, the “output_conv” block was extended by increasing the convolutional layers from 2 to 4. Each layer now receives an additional input tensor “rho_input” of size 128×128 that contains the density value.

The output of the model is a two-dimensional array of size 128×128. The values of the previously generated solutions were normalized to a distribution with a mean value of 0 and a standard deviation of 1 to ensure the stability of the learning of the convolutional network. The process of obtaining the solution of the Poisson equation in physical units of measurement is as follows: the value of the output array of the neural network is inversely normalized, and thus the original distribution of the training data is restored.

To create a training dataset that will ensure the accuracy and efficiency of the neural network, 18 space geometries were prepared. They were used to simulate the movement of liquids using the LBM method. The numerical solution of equation (12) was used to calculate the values of the pressure field. Different random values of the initial fluid velocity, density, and relaxation time were used in the simulation of each geometry to ensure the variability of the dataset. In this case, the neural network features are the values of free members and density, and the target variable is the pressure value. In this way, a training dataset consisting of 75,000 objects was formed. A test dataset of 10,000 objects was created in the same way.

To implement the developed neural network, we used the following software: Python programming language, TensorFlow 2.4.1 machine learning framework. The Adam optimizer [22] was used to optimize the parameters of the neural network. The values of the optimizer parameters were as follows: learning_rate=0.0005, beta_1 = 0.95, beta_2 = 0.99, epsilon = 1e–7. Each of the parameters performs the following function: learning_rate is a parameter in the optimization algorithm that determines the step size of updating the coefficients of the neural network at each learning iteration; beta_1 is the forgetting

© Novotarskyi M. A., Kuzmych V. A., 2023
DOI 10.15588/1607-3274-2023-4-6

coefficient for the gradient; beta_2 is the forgetting factor for the second moment of the gradient; epsilon is a small constant introduced to ensure optimization stability. The average absolute error represents the loss function for this model. An MSI GeForce GTX 1660 Super Ventus OC 6GB GDDR6 GPU was used to train the neural network for 300 epochs. The value of the average absolute error on the test data set was 0.001021.

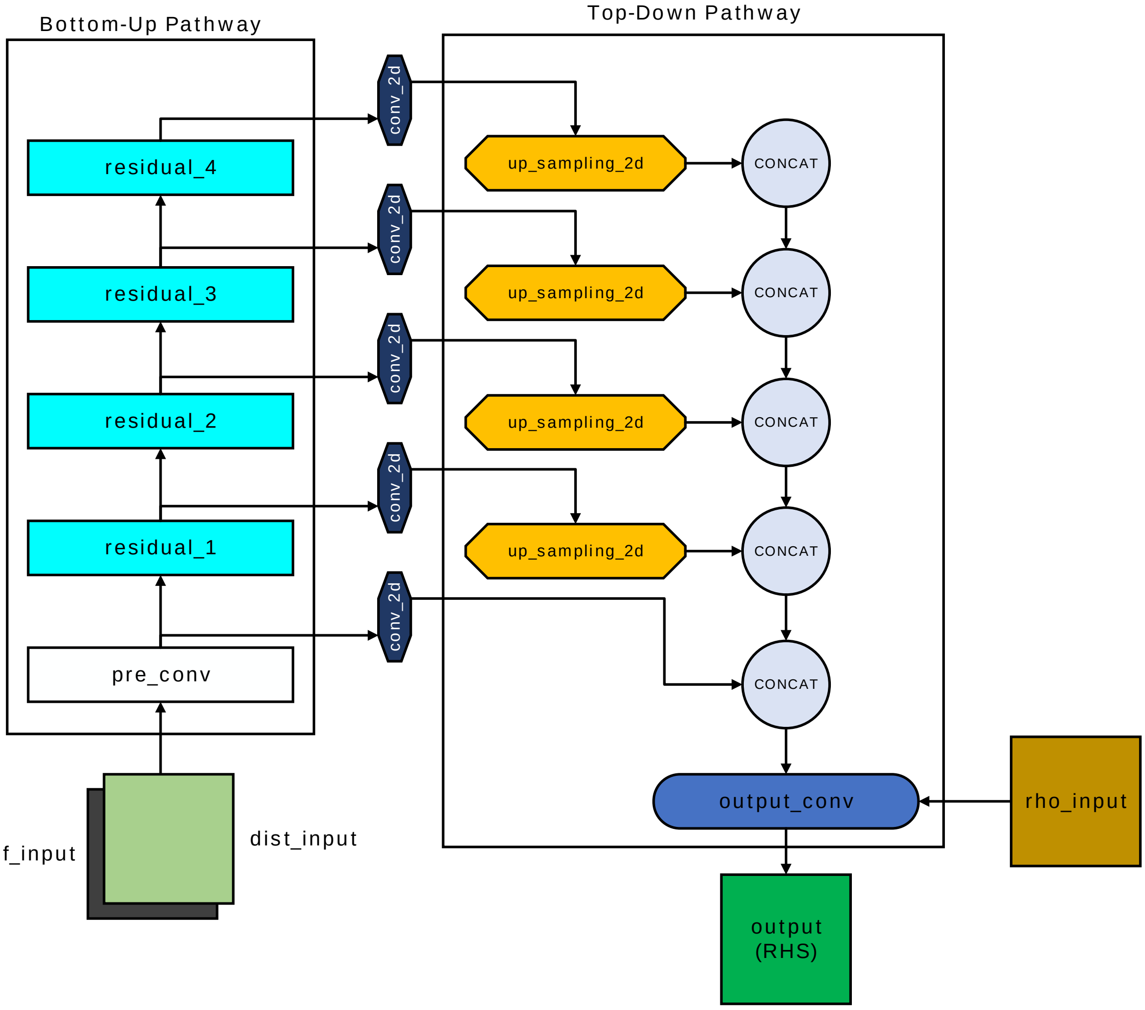

Figure 3 – The general structure of a convolutional neural network

4 EXPERIMENTS

To test the developed method, we used the model of the section of the human colon, which is shown in Fig. 4.

Black color indicates the working area, white color indicates the bounding surface.

The working area was discretized by a 128×128 grid. The parameters of the method are as follows: $\tau = 0.5012$, $\rho = 1000$. The boundary condition of fluid inflow from the right boundary was applied, equal to $v_0 = 0.01 \times \left| \sin\left(\frac{2\pi t}{T} \right) \right|$, where $T = 110$.

In the experiment, modeling was carried out using two methods: LBM and the LBM method with velocity field correction. The velocity field was measured at simulation iterations 300 and 900. The simulation results are shown in Fig. 5 and 6.

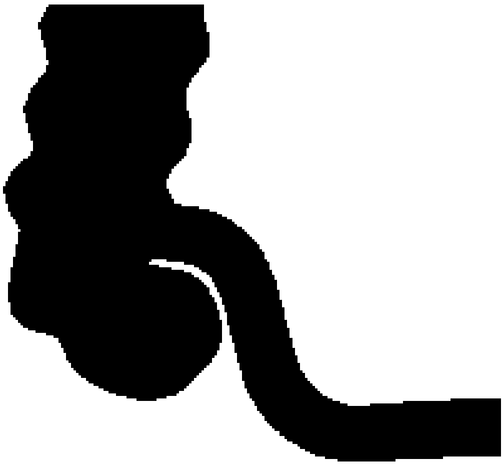

Figure 4 – Model of the section of the human colon

DOI 10.15588/1607-3274-2023-4-6

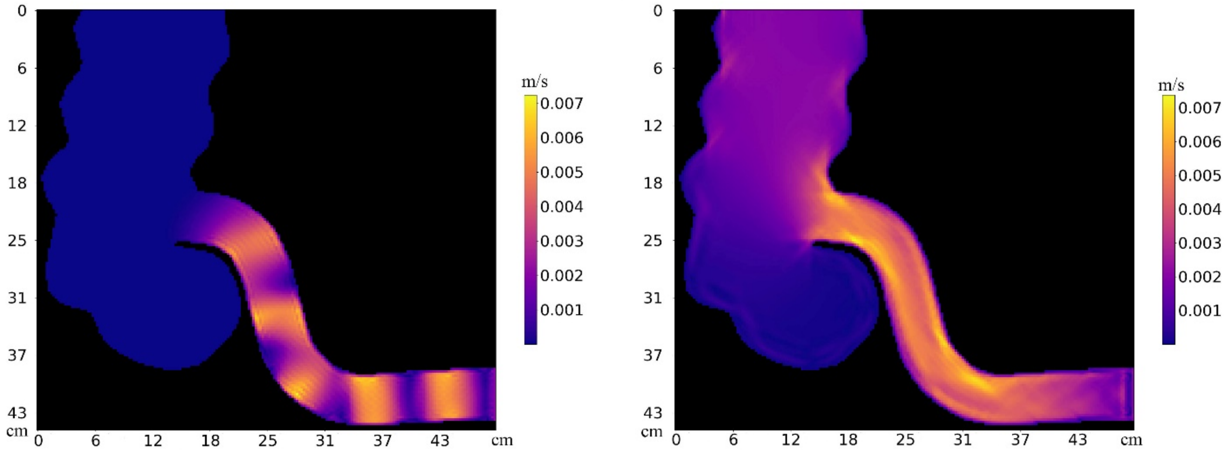

Figure 5 – Distribution of the velocity field at the 300th iteration. On the left is the LBM method, on the right is the proposed method with speed correction

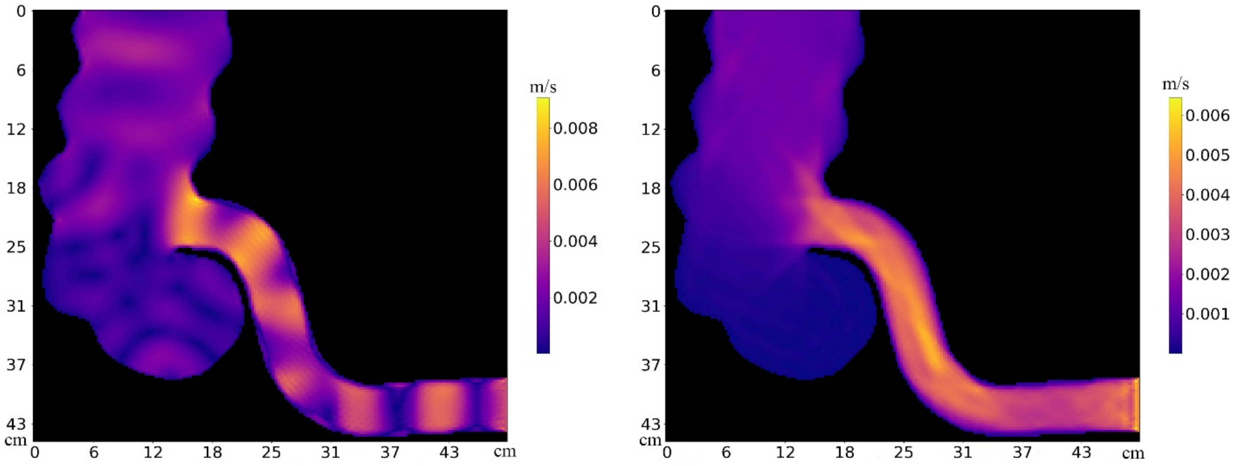

Figure 6 – Distribution of the velocity field at the 900th iteration. On the left is the LBM method, on the right is the proposed method with speed correction

Fig. 7 shows the results of measurements of the deviation of the average density values and from the initial density value in the computational domain after 1500 iterations.

When modeling with the developed method, the density values deviate less from the initial density value than the usual LBM method. Therefore, the obtained results indicate higher stability when modeling fluid movement using the combined method.

The computational speed of the developed method was measured in comparison with other methods. Three methods were used for comparison. The first method is the conventional LBM method without rate correction described in equation (14), the second method is the proposed rate-corrected LBM method using a convolutional neural network to solve equation (13), the third method is a velocity-corrected LBM method that uses the AMG numerical method [23] to solve equation (13). The comparative results of the experiments are shown in Table 1.

Table 1 – Comparison of modeling methods

Modeling method	**Time of one iteration, sec**
LBM	0.00301
LBM+ neural network	0.06087
LBM+ numerical method	0.54123

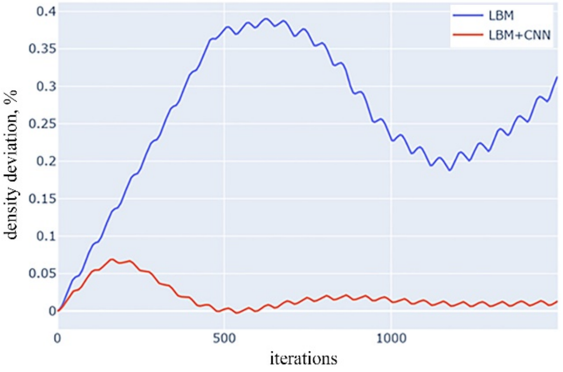

Figure 7 – The average value of the deviation of the density during the simulation

DOI 10.15588/1607-3274-2023-4-6

The experimental results show that the simulation speed of one iteration of the developed method is 6–7 times faster than the speed of the method that uses the numerical solution of the Poisson equation. Also, in comparison with the usual LBM method, the condition of fluid incompressibility is preserved.

5 RESULTS

The main result is that a new method of modeling two-dimensional hydrodynamic processes is proposed, which is characterized by increased accuracy and lower resource consumption compared to known methods. The sequence of actions for implementing the method is represented by the algorithm shown in Fig 1. The positive effect is achieved by combining the modified lattice Boltzmann method to determine the velocity field at the next time step. At each time step, we simulate the solution of the Poisson equation using a convolutional neural network to determine the pressure distribution. The obtained results make it possible to correct the deviation of the density to increase the simulation's accuracy. A parallel numerical method was used to reduce the time of preparing a dataset for neural network training. To practically confirm this approach's effectiveness, modeling the physical process of fluid movement in a fragment of the digestive tract was performed. All technical parameters of this experiment are given in this paper. Studies have shown that the simulation speed of one iteration of the developed method is 6–7 times faster than the speed of the method that uses the numerical solution of the Poisson equation. The value of the average absolute error on the set of test data is 0.001021.

6 DISCUSSION

The traditional approach to modeling hydrodynamic processes is based on solving a boundary value problem, which includes the Navier-Stokes equation and the flow continuity equation. This problem can be solved by one of the numerical methods, which causes significant difficulties in the case of the complex geometry of the surface area and requires significant resources since it is necessary to use excessive accuracy to ensure the convergence of the numerical method.

When modeling the movement of liquids in living nature, it is essential to get an estimate of the dynamics of the process in a short time with relatively low accuracy. Therefore, LBM is more often used for such tasks. This method allows us to get the desired solution in a significantly shorter time, but several disadvantages characterize it. One of the critical disadvantages is that this method is fundamentally oriented towards compressible liquids, and the dynamics of density change are challenging to control. Another significant drawback is the considerable laboriousness of the correct definition of the boundary conditions, given the complex geometry of the area.

Another approach to solving this problem includes using a predictor-corrector method. The predictor applies an explicit iterative scheme to determine the velocity field at the next time layer, and the corrector refines the value of the velocity field using the pressure distribution.

This work is devoted to developing this approach by improving both the predictor and corrector stages. To improve the accuracy of the simulation, a modified LBM is used in the predictor stage, and a pre-trained convolutional neural network is applied in the corrector stage to solve the Poisson equation. Combining these two approaches made it possible to improve the speed and accuracy of modeling in a region with complex geometry.

Further development of this approach consists of applying reinforcement learning mechanisms at the corrector stage, which will improve the accuracy of the obtained modeling results with significant changes in the geometry of the area without spending time retraining the convolutional network.

After certain refinements, the proposed method can be used to study hydrodynamic parameters in three-dimensional domains.

CONCLUSIONS

The urgent problem of developing a software system for mathematical modeling of the movement of liquids in areas with complex geometry, which is characteristic of living organisms, is being solved.

The scientific novelty of the obtained results lies in the fact that, for the first time, a method for modeling the movement of fluids in living organisms has been proposed, which reduces the modeling time and increases its accuracy compared to known approaches due to the synergistic effect obtained by improving the modeling parameters at each step of the iterative process, which includes a predictor stage and a corrector stage. The predictor stage implements a modified LBM in which a modified equilibrium distribution function is applied, which increases the accuracy of determining the distribution function in one iteration step by the BGK model. The LBM implementation time is reduced by parallelizing the calculation of discrete values of the distribution function during the collision of elementary liquid volumes at the mesoscopic level. The corrector stage realizes a reduction in modeling time due to the use of a previously trained convolutional network, the structure of which is adapted to the solution of a specific problem.

The practical significance of the obtained results is that a software system has been developed for simulating the movement of liquids in areas of complex shape, significantly reducing the simulation time and using computing resources, provided that the obtained results are of acceptable accuracy. Experiments were conducted using different workloads of the developed software simulation system. The obtained results made it possible to recommend using this software system when studying the movement of fluids in living nature, particularly when studying the digestive and cardiovascular systems of living organisms.

Prospects for further research consist in expanding the possibilities of this method for its application to three-dimensional areas of complex shape. Another direction

DOI 10.15588/1607-3274-2023-4-6

for improving this method is using reinforcement learning mechanisms to reduce the time it takes to reconfigure a convolutional neural network for an expanded range of tasks.

ACKNOWLEDGEMENTS

The work was supported by a NATO grant (program: “Science for peace and security”). State registration number 0123U101109 dated 02/14/2023.

REFERENCES

1. Chen S., Doolen G. D. Lattice Boltzmann method for fluid flows, *Annual Review of Fluid Mechanics,* 1998, Vol. 30, pp. 329–364. https://doi.org/10.1146/annurev.fluid.30.1.329
2. Guo Z., Shu Ch. Lattice Boltzmann Method and Its Applications in Engineering. Singapore, World Scientific, 2013, 420 p. https://doi.org/10.1142/8806
3. Tessarotto M., Fonda E., Tessarotto M. The computational complexity of traditional Lattice Boltzmann methods for incompressible fluids, *Rarefied Gas Dynamics: The 26th International Symposium, Kyoto (Japan), 20–25 July 2008 : proceedings*. New York, AIP Publishing, AIP Conference Proceedings, 2008, Vol. 1084, № 1, pp. 470–475, https://doi.org/10.1063/1.3076524
4. Zimny S., Masilamani K., Jain K., Roller S. P. Lattice Biltzmann Simulations on Complex Geometries [Electronic resource]. Access mode: https://www.researchgate.net/publication/289041300_Latice _Boltzmann_Simulations_on_Complex_Geometries
5. Korner C., Pohl T., Rude U., Thürey N., Zeiser T. Parallel Lattice Boltzmann Method for CFD Applications, *Lecture Notes in Computational Science and Engineering*. Berlin, Springer, 2006, Vol. 51, pp.439–466, https://doi.org/10.1007/3-540-31619-1_13
6. Xunliang L., Wen-Quan T., He Y. L. A simple method for improving the SIMPLER algorithm for numerical simulations of incompressible fluid flow and heat transfer problems, *Engineering Computations*. Berlin, Springer, 2005, Vol.22, pp. 921–939. https://doi.org/10.1108/02644400510626488
7. Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators, *Neural Networks*, 1989, Vol. 2, №5, pp. 359–366, https://doi.org/10.1016/0893-6080(89)90020-8
8. Lagaris I. E., Likas A. C., Papageorgiou D. G. Neural-network methods for boundary value problems with irregular boundaries, *IEEE Transfctions on Neural Networks*, 2000, Vol. 11, №5, pp. 1041–1049, https://doi.org/10.1109/72.870037
9. Aarts L. P., Veer P. V. Neural network method for solving partial differential equations, *Neural Processing Letters.* Berlin, Springer, 2001, Vol. 14, pp. 261–271.
10. Smaoui N., Al-Enezi S. Modeling the dynamics of nonlinear partial differential equations using neural networks, *Journal of Computational and Appliaed Mathematics*, 2004, Vol. 170, pp. 27–58, https://doi.org/10.1016/j.cam.2003.12.045
11. Deheuvels P., Martynov G. V. A Karhunen-Loeve decomposition of a Gaussian process generated by independent pairs of exponential random variables, *Journal of Functional Analysis*, 2008, Vol. 255, pp. 2363–2394
12. Chua L.O., Yang L. Cellular neural networks: theory, *IEEE Transactions on Circuits Systems,* 1988, Vol. 35, № 10, pp. 1257–1272. https://doi.org/10.1109/31.7600
13. Novotars'kij M. A., Nesterenko B. B. Shtuchni nejronni merezhi: obchis-lennya. Kyiv, In-t matematiki, 2004, 408 p.
14. He X., Luo L.-S. A Priori Derivation of the Lattice Boltzmann Equation, *Physical Review E,* 1997, Vol. 55, № 6, pp. R6333–R6336.
15. Eichinger M., Heinlein A., Klawonn A. Stationary flow predictions using convolutional neural networks, *Numerical Mathematics and Advanced Applications.* Berlin, Springer, 2021. – P.541–549
16. Koelman J.M.V.A. A simple lattice Boltzmann scheme for Navier-Stokes fluid flow, *Europhysics Letters,* 1991, Vol. 15, № 6, pp. 603–60. https://doi.org/10.1209/0295-5075/15/6/007
17. Qian Y. H., d’Humieres D., Lallemand P.] Lattice BGK models for Navier-Stokes equation, *Europhysics Letters,* 1992, Vol. 17, № 6, pp. 479–484, https://doi.org/10.1209/0295-5075/17/6/001
18. Sterling D. J., Chen S. Stability Analysis of Lattice Boltzmann Methods, *Journal of Computational Physics,* 1996, Vol. 123, № 1, pp. 196–206, https://doi.org/10.1006/jcph.1996.0016
19. Chorin A. J. Numerical solution of the Navier-Stokes equations, *Mathematics of computation*, 1968, Vol. 22, № 104, pp. 745–762. https://doi.org/10.2307/2004575
20. Vittoz G., Oger M., de Leffe M., Le Touze D.] Comparisons of weakly-compressible and truly incompressible approaches for viscous flow into a high-order Cartesian-grid finite volume framework, *Journal of Computational Physics*, 2019, Vol. 1, pp. 100015–100048. https://doi.org/10.1016/j.jcpx.2019.100015
21. Kuzmych V., Novotarskyi M. Accelerating simulation of the PDE solution by the structure of the convolutional neural network modifying : Lecture Notes on Data Engineering and Communications, *Artificial Intelligence and Logistics Engineering (ICAILE2022): The 2nd International Conference, 20–22 February 2022 : proceedings*. Berlin, Springer, 2022, pp. 3–15. https://doi.org/10.1007/978-3-031-04809-8_1
22. Duchi J., Hazan E., Singer Y. Adaptive subgradient methods for online learning and stochastic optimization, *Journal of Machine Learning Research*, 2011, Vol. 12, pp. 2121–2159
23. Brandt A., McCormick S., Ruge J. Algebraic multigrid (AMG) for sparse matrix equations [Electronic resource]. Access mode: https://www.researchgate.net/publication/243769570_Algeb raic _multigrid_AMG_for_sparse_matrix_ equations

Received 16.10.2023.
Accepted 27.11.2023.

DOI 10.15588/1607-3274-2023-4-6

УДК 004.942:001.57

МЕТОД ГІДРОДИНАМІЧНОГО МОДЕЛЮВАННЯ З ВИКОРИСТАННЯМ ЗВЕРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ

Новотарський М. А. – д-р техн. наук, професор кафедри обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

Кузьмич В. А. – аспірант кафедри обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

АНОТАЦІЯ

Актуальність. Розв'язування гідродинамічних задач пов'язане з високою обчислювальною складністю і тому вимагає значних обчислювальних ресурсів і часу. Запропонований підхід дозволяє суттєво скоротити час розв'язування таких задач шляхом застосування комбінації двох вдосконалених методів моделювання.

Мета. Метою є створення комплексного методу гідродинамічного моделювання, який вимагає значно менше часу для визначення динаміки поля швидкостей за рахунок використання модифікованого решітчастого методу Больцмана і розподілу тиску за рахунок використання згорткової нейронної мережі.

Метод. Запропоновано метод гідродинамічного моделювання, який реалізує синергетичний ефект, що виникає при поєднанні вдосконаленого решітчастого методу Больцмана та згорткової нейронної мережі з спеціально адаптованою структурою. Суть методу полягає у реалізації послідовності ітерацій, на кожній з яких відбувається моделювання процесу зміни параметрів при переході на наступний часовий шар. Кожна ітерація включає крок предиктора та крок коректора. На кроці предиктора працює решітчастий метод Больцмана, який дозволяє отримати поле швидкостей рідини в робочій зоні на наступному часовому шарі за допомогою поля швидкостей на попередньому шарі. На кроці коректора ми застосовуємо вдосконалену згорткову нейронну мережу, навчену на раніше створеному наборі даних. Використання нейронної мережі дозволяє визначити розподіл тиску на новому часовому шарі із заданою точністю. Після додавання поправки на стисливість рідини на новому часовому шарі ми отримуємо уточнені значення поля швидкостей, які можна використовувати як початкові дані для застосування решітчастого методу Больцмана на наступній ітерації. Обчислення припиняються при досягненні заданої кількості ітерацій.

Результати. Роботу запропонованого методу досліджено на прикладі моделювання руху рідини у фрагменті шлунково-кишкового тракту людини. Результати моделювання показали, що час, витрачений на реалізацію процесу моделювання, скоротився у 6–7 разів при збереженні прийнятної для практичних завдань точності.

Висновки. Запропонований метод гідродинамічного моделювання зі згортковою нейронною мережею та решітчастим методом Больцмана суттєво скорочує час та обчислювальні ресурси, необхідні для реалізації процесу моделювання в областях зі складною геометрією. Подальший розвиток цього методу дозволить реалізувати гідродинамічне моделювання в реальному часі в тривимірних областях.

КЛЮЧОВІ СЛОВА: гідродинамічне моделювання, згорткова нейронна мережа, решітчастий метод Больцмана.

ЛІТЕРАТУРА

1. Chen S. Lattice Boltzmann method for fluid flows/ S. Chen, G. D. Doolen // Annual Review of Fluid Mechanics. – 1998. – Vol. 30. – P. 329–364, https://doi.org/10.1146/annurev.fluid.30.1.329
2. Guo Z. Lattice Boltzmann Method and Its Applications in Engineering / Z. Guo, Ch. Shu. – Singapore : World Scientific, 2013. – 420 p. https://doi.org/10.1142/8806
3. Tessarotto M. The computational complexity of traditional Lattice Boltzmann methods for incompressible fluids / M. Tessarotto, E. Fonda, M. Tessarotto // Rarefied Gas Dynamics: The 26th International Symposium, Kyoto (Japan), 20–25 July 2008 : proceedings. – New York: AIP Publishing, AIP Conference Proceedings, 2008. – Vol. 1084, № 1. – P.470–475, https://doi.org/10.1063/1.3076524
4. Lattice Biltzmann Simulations on Complex Geometries [Electronic resource] / [S. Zimny, K. Masilamani, K. Jain, S. P. Roller] . – Access mode: https://www.researchgate.net/publication/289041300_Latice_Boltzmann_Simulations_on_Complex_Geometries
5. Parallel Lattice Boltzmann Method for CFD Applications / [C. Korner, T. Pohl, U. Rude et al.] // Lecture Notes in Computational Science and Engineering. – Berlin : Springer, 2006. – Vol. 51. – P.439–466, https://doi.org/10.1007/3-540-31619-1_13
6. Xunliang L. A simple method for improving the SIMPLER algorithm for numerical simulations of incompressible fluid flow and heat transfer problems / L. Xunliang, T. Wen-Quan, Y. L. He // Engineering Computations. – Berlin : Springer. – 2005 . – Vol. 22. – P. 921–939. https://doi.org/10.1108/02644400510626488
7. Hornik K. Multilayer feedforward networks are universal approximators / K. Hornik, M. Stinchcombe, H. White // Neural Networks. – 1989. – Vol. 2, № 5. – P. 359–366, https://doi.org/10.1016/0893-6080(89)90020-8
8. Lagaris I. E. Neural-network methods for boundary value problems with irregular boundaries / I. E. Lagaris, A. C. Likas, D. G. Papageorgiou // IEEE Transfctions on Neural Networks. – 2000. – Vol. 11, № 5. – P. 1041–1049, https://doi.org/10.1109/72.870037
9. Aarts L. P. Neural network method for solving partial differential equations / L. P. Aarts, P. V. Veer // Neural Processing Letters. – Berlin : Springer, 2001. – Vol. 14. – P. 261–271.
10. Smaoui N. Modeling the dynamics of nonlinear partial differential equations using neural networks / N. Smaoui, S. Al-Enezi // Journal of Computational and Appliaed Mathematics. – 2004. – Vol. 170. – P. 27–58, https://doi.org/10.1016/j.cam.2003.12.045
11. Deheuvels P. A Karhunen-Loeve decomposition of a Gaussian process generated by independent pairs of exponential random variables / P. Deheuvels, G. V. Martynov // Journal of Functional Analysis. – 2008. – Vol. 255. – P. 2363–2394.
12. Chua L. O. Cellular neural networks: theory / L. O. Chua, L. Yang // IEEE Transactions on Circuits Systems. – 1988. –

DOI 10.15588/1607-3274-2023-4-6

Vol. 35, № 10. – P. 1257–1272. https://doi.org/10.1109/31.7600

13. Новотарський М. А. Штучні нейронні мережі: обчислення / М. А. Новотарський, Б. Б. Нестеренко. – Київ: Ін-т математики, 2004. – 408 с.
14. He X. A Priori Derivation of the Lattice Boltzmann Equation / X. He, L.-S. Luo // Physical Review E. – 1997. – Vol. 55, № 6. – P. R6333–R6336.
15. Stationary flow predictions using convolutional neural networks / [M. Eichinger, A. Heinlein, A. Klawonn] // Numerical Mathematics and Advanced Applications. – Berlin: Springer. – 2021. – P.541–549
16. Koelman J.M.V.A. A simple lattice Boltzmann scheme for Navier-Stokes fluid flow / J.M.V.A. Koelman // Europhysics Letters. – 1991. – Vol. 15, № 6. – P. 603–607. https://doi.org/10.1209/0295-5075/15/6/007
17. Qian Y. H. Lattice BGK models for Navier-Stokes equation / [Y. H. Qian, D. d'Humieres, P. Lallemand] // Europhysics Letters. – 1992. – Vol. 17, №6. – P. 479–484, https://doi.org/10.1209/0295-5075/17/6/001
18. Sterling D. J. Stability Analysis of Lattice Boltzmann Methods / D. J. Sterling, S. Chen // Journal of Computational Physics. – 1996. – Vol. 123, № 1. – P. 196–206, https://doi.org/10.1006/jcph.1996.0016
19. Chorin A. J. Numerical solution of the Navier-Stokes equations / A. J. Chorin // Mathematics of computation. – 1968. – Vol. 22, № 104. – P.745–762. https://doi.org/10.2307/2004575
20. Comparisons of weakly-compressible and truly incompressible approaches for viscous flow into a high-order Cartesian-grid finite volume framework / [G. Vittoz, M. Oger, M. de Leffe, D. Le Touze] // Journal of Computational Physics. – 2019. – Vol. 1. – P. 100015–100048. https://doi.org/10.1016/j.jcpx.2019.100015
21. Kuzmych V. Accelerating simulation of the PDE solution by the structure of the convolutional neural network modifying : Lecture Notes on Data Engineering and Communications / V. Kuzmych, M. Novotarskyi // Artificial Intelligence and Logistics Engineering (ICAILE2022): The 2nd International Conference, 20–22 February 2022 : proceedings. – Berlin: Springer, 2022. – P. 3–15, https://doi.org/10.1007/978-3-031-04809-8_1
22. Duchi J. Adaptive subgradient methods for online learning and stochastic optimization / J. Duchi, E. Hazan, Y. Singer // Journal of Machine Learning Research, 2011. – Vol. 12. – P. 2121–2159.
23. Brandt A. Algebraic multigrid (AMG) for sparse matrix equations [Electronic resource] / A. Brandt, S. McCormick, J. Ruge. – Access mode: https://www.researchgate.net/publication/243769570_Algebraic_multigrid_AMG_for_sparse_matrix_ equations

DOI 10.15588/1607-3274-2023-4-6